EXHIBIT I

U.S. PATENT NO. 5,630,096

Claim	Anticipation/Obviousness
1. A controller for a synchronous DRAM comprising:	MT48LC2M8S1 The Micron MT48LC2M8S1 (S) 2 MEG x 8 SDRAM Advance Data Sheet (April 1994) ("Micron Data Sheet") discloses a synchronous DRAM. U.S. Patent No. 5,896,551 to Williams (Apr. 20, 1999) ("Williams '551") discloses a synchronous DRAM. *See, e.g.*, Williams '551 at 1:26-36. U.S. Patent No. 5,717,639 to Williams (Feb. 10, 1998) ("Williams '639") discloses a synchronous DRAM. *See, e.g.*, Williams '639 at 1:27-37. U.S. Patent No. 5,887,162 to Williams (Mar. 23, 1999) ("Williams '162") discloses a synchronous DRAM. *See, e.g.*, Williams '162 at 1:20-30. U.S. Patent No. 6,175,901 to Williams (Jan. 16, 2001) ("Williams '901"). *See, e.g.*, Williams '901 at 1:26-36. 82C302 The 82C302 Page/Interleave Memory Controller ("the 82C302") is a DRAM controller manufactured by Chips & Technologies, Inc. *AT386 CHIPSet Functional Specification*, Chips and Technologies, Inc. (May 8, 1986) ("*AT386 Specification*") discloses a DRAM controller. *See, e.g.*, AT386 Specification at 2, §§ 6.0-6.1. *82C302 Page/Interleave Memory Controller Data Sheet*, Chips and Technologies, Inc. (1987) ("*82C302 Data Sheet*") discloses a DRAM controller. *See, e.g.*, *82C302 Data Sheet* at 43-65. MPC105 *MPC105 PCI Bridge/Memory Controller Technical Summary*, Motorola, Inc. (Jan. 1995) ("*MPC105 Summary*") discloses a controller for a synchronous DRAM. Karl Wang *et al*, "Designing the MPC105 PCI Bridge/Memory Controller," *IEEE Micro* 44-49 (Apr. 1995) ("Wang") discloses a controller for a synchronous DRAM. Michael J. Garcia & Brian K. Reynolds, "Single Chip PCI Bridge and Memory Controller for PowerPC™ Microprocessors," *IEEE International Conference on Computer Design: VLSI in Computers and Processors* 409-12 (Oct. 1994) ("Garcia") discloses a controller for a synchronous DRAM. Stream Memory Controller The Stream Memory Controller is a DRAM controller. U.S. Patent No. 6,154, 826 to Wulf (Nov. 28, 2000) ("Wulf") discloses

	a DRAM controller. Sally A. McKee, "Hardware Support for Dynamic Access Ordering: Performance of Some Design Options," Computer Science Report No. CS-93-08 (Aug. 9, 1993) ("McKee I") discloses a DRAM controller. *See, e.g.*, McKee I at 4-6. Sally A. McKee *et al.*, "Experimental Implementation of Dynamic Access Ordering," *Proceedings of the Twenty-Seventh Hawaii International Conference* (Jan. 1994), Computer Science Report No. CS-93-42 (Aug. 1, 1993) ("McKee II") discloses a DRAM controller. *See, e.g.*, McKee II at 1-4, Figs. 3, 9-10. Sean W. McGee *et al.*, "Design of a Processor Bus Interface ASIC for the Stream Memory Controller," *Proc. IEEE Int'l ASIC Conference*, Rochester, NY (Sep. 1994) ("McGee") discloses a DRAM controller. *See, e.g.*, McGee at 462-63. CS4031 The 84031 is a DRAM controller manufactured by Chips and Technologies, Inc. *CS4031 CHIPSet Advance Product Information* (May 10, 1993) ("*CS4031 Information*") discloses a DRAM controller. *See, e.g.*, *CS4031 Information* at 1, 3, 80. CDC-6600 The CDC-6600 is a memory controller for synchronous memory. J.E. Thornton, *Design of a Computer: The Control Data 6600* (1970) ("Thornton") describes the CDC-6600 and discloses a memory controller for synchronous memory. *See, e.g.*, Thornton at 47-48, Fig. 35. The CDC-6600 is covered in computer science textbooks and was well known in the art at the time of the alleged invention of the '096 Patent. One of ordinary skill in the art would have been motivated to adapt the synchronous memory controller of the CDC-6600 for use with DRAM. JOHNSON/AMD TECHNOLOGY U.S. Patent No. 5,805,912 to Johnson (Sep. 8, 1998) ("Johnson '912") discloses a DRAM controller. *See, e.g.*, Johnson '912 at 1:33-43, 2:56-67, 5:43-67, 6:17-67. U.S. Patent No. 5,828,869 to Johnson (Oct. 27, 1998) ("Johnson '869") discloses a DRAM controller. *See, e.g.*, Johnson '869 at 1:33-43, 2:57-67, 5:43:67, 6:17-67. OTHER REFERENCES R.E. Busch *et al.*, "Dynamic Random Access Memory Control Data Burst," *IBM Technical Disclosure Bulletin* vol. 37, no. 9 (Sept. 1994) ("Busch") discloses a controller for a synchronous DRAM. *See, e.g.*, Busch at 649 ("A method to control sequential or interleaved data being

	written into or read from Dynamic Random Access Memory (DRAM) modules is disclosed."), 649-50 ("Another improvement in data rate is achieved with Synchronous DRAM or SDRAM, where data is transfered [sic] at the rising edge of a continuously running clock signal."), 652. U.S. Patent No. 5,577,236 to Johnson (Nov. 19, 1996) ("Johnson '236") discloses a controller for a synchronous DRAM. *See, e.g.*, Johnson '236 at 1:40-2:5, 5:20-33, claims 1-17, Fig. 3, Abstract. Dave Bursky, *Sychrounous DRAMs Clock at 100MHz*, Electronic Design 45-48 (Feb. 18, 1993) ("Bursky") discloses a controller for a synchronous DRAM. *See, e.g.*, Bursky at 46. U.S. Patent No. 4,803,621 to Kelly (Feb. 7, 1989) ("Kelly") discloses a DRAM controller. *See* Kelly, Fig. 2. U.S. Patent No. 6,807,609 to Lemmon (Oct. 19, 2004) ("Lemmon II") discloses a DRAM controller. *See, e.g.*, Lemmon II at 2:5-9. U.S. Patent No. 5,278,974 to Lemmon (Jan. 11, 1994) ("Lemmon I") discloses a DRAM controller. *See* Lemmon I at 5:25-8:29, Fig. 3. U.S. Patent No. 5,701,434 to Nakagawa (Dec. 23, 1997) ("Nakagawa") discloses a DRAM controller. *See, e.g.*, Nakagawa at 2:54-58, 6:3-4, Figs. 1-3, claims 1, 4. U.S. Patent 5,208,914 to Wilson (May 4, 1993) ("Wilson") discloses a memory controller. *See, e.g., Wilson*, 1:25-27, 2:46-52, 3:3-23, 4:5-10, 4:56-65, 12:50-55, Figs. 2, 3a, 3b. U.S. Patent 5,197,130 to Chen (Mar. 23, 1993) ("Chen") discloses a memory controller. *See, e.g.*, Chen at 14:40-54, Figs. 10, 14. U.S. Patent 5,168,547 to Miller (Dec. 1, 1992) ("Miller") discloses a memory controller. *See, e.g.*, Miller, Fig. 1. U.S. Patent 4,527,232 to Bechtolsheim (Jul. 2, 1985) ("Bechtolsheim") discloses a memory controller. *See, e.g.*, Bechtolsheim at 1:60-2:18, claims 1-5. U.S. Patent No. 5,513,327 to Farmwald (Apr. 30, 1996) ("Farmwald '327") discloses a DRAM controller. *See, e.g.*, Farmwald '327 at 3:53-4:47, Fig. 2. International Patent Application No. WO 91/16680 (Oct. 31, 1991) ("WO 91/16680") discloses a DRAM controller. *See, e.g.*, WO 91/16680 at 7:20-8:8, Fig. 2. U.S. Patent No. 5,511,024 to Ware (Apr. 23, 1996) ("Ware '024") discloses a controller for a synchronous DRAM. *See, e.g.*, Ware '024 at 1:39-45. D.J. Lang *et al.*, "Enhanced Refresh Mechanism for Higher

	Performance in Memory Subsystems," *IBM Technical Disclosure Bulletin* vol. 37, no. 10 (Oct. 1994) ("Lang") discloses a DRAM controller. *See* Lang at 483, Fig. 1. U.S. Patent No. 5,179,667 to Iyer (Jan. 12, 1993) ("Iyer") discloses a DRAM controller. *See e.g.*, Iyer at 1:1-2. U.S. Patent No. 5,327,570 to Foster (Jul. 5, 1994) ("Foster") discloses a DRAM controller. *See, e.g.*, Foster at 4:61-66. U.S. Patent No. 4,796,232 to House (Jan. 3, 1989) ("House") discloses a DRAM controller. *See, e.g.*, House at 2:45-51, 3:29-4:5, 4:39-63, 5:28-6:25, Figs. 1, 7, Abstract. M.J. Carnevale *et al.*, "Fast Data Access of DRAMs by Utilizing a Queued Memory Command Buffer," *IBM Technical Disclosure Bulletin* vol. 35, no. 7 (Dec. 1992) ("Carnevale") discloses a DRAM controller. Carnevale at 63-66. U.S. Patent No. 5,732,236 to Nguyen (Mar. 24, 1998) ("Nguyen") discloses a DRAM controller. *See, e.g.*, Nguyen at 1:8-17, 1:64-2:22, 2:58-3:3, 4:20-27, Figs. 1-2, Abstract. U.S. Patent No. 5,638,534 to Mote (Jun. 10, 1997) ("Mote '534) discloses a DRAM controller. *See, e.g.*, Mote '534 at 1:9-40, 2:66-3:3, 5:51-54, 6:25-38,10:11-67, 19:39-45, 20:32-41, 22:1-11, Figs. 1-2, 7-8. U.S. Patent No. 5,666,494 to Mote (Sep. 9, 1997) ("Mote '494) discloses a DRAM controller. *See, e.g.*, Mote '534 at 1:9-40, 2:66-3:3, 5:51-54, 6:25-38,10:11-67, 19:39-45, 20:32-41, 22:1-11, Figs. 1-2, 7-8; Mote '494 at 1:8-39, 2:65-3:2, 5:51-54, 6:25-38,10:11-67, 20:24-27, Figs. 1-2, 7-8. U.S. Patent No. 4,843,543 to Isobe (Jun. 27, 1989) ("Isobe") discloses a memory controller. *See, e.g.*, Isobe at 1:7-11, 4:56-5:7, 6:44-51, Figs. 1-3. U.S. Patent No. 4,937,791 to Steele (Jun. 26, 1990) ("Steele") discloses a DRAM controller. *See, e.g.*, Steele at 1:6-24, 4:60-7:59, Figs. 6, 8, 12, Abstract. U.S. Patent No. 5,034,917 to Bland (Jul. 23, 1991) ("Bland") discloses a DRAM controller. *See, e.g.*, Bland at 1:16-4:40. U.S. Patent No. 5,303,364 to Mayer (Apr. 12, 1994) ("Mayer U.S.") discloses a DRAM controller. *See, e.g.*, Mayer U.S. at 1:9-12, 4:18-30, claims 1, 8-11, 15-20, Figs. 1-2, Abstract. Published European Patent Application No. EP0427425A2 to Mayer (May 15, 1991) ("Mayer Euro") discloses a DRAM controller. *See, e.g.*, Mayer Euro at [57], 1:1-4, 4:56-5:11, claims 1, 3, 5, 7, 10-14 Figs. 1-2. U.S. Patent No. 5,440,713 to Lin (Aug. 8, 1995) ("Lin") discloses a

	DRAM controller. *See, e.g.*, Lin at 1:19-32, 2:53-63, 3:35-37, 6:25-68. U.S. Patent No. 5,301,278 to Bowater (Apr. 5, 1994) ("Bowater") discloses a DRAM controller. *See, e.g.*, Bowater at 1:11-3:20, 3:43-5:19, 12:52-13:11, Figs. 3-4. U.S. Patent No. 5,615,355 to Wagner (Mar. 25, 1997) ("Wagner") discloses a DRAM controller. *See, e.g.*, Wagner at 1:58-62, Figs. 3-4, Abstract. U.S. Patent No. 5,812,829 to Ito (Sep. 22, 1998) ("Ito") discloses a controller for a synchronous DRAM. *See, e.g.*, Ito at 2:54-4:33, 7:43-62, Fig. 1. U.S. Patent No. 6,008,850 to Sumihiro (Dec. 28, 1999) ("Sumihiro U.S.") discloses a controller for a synchronous DRAM. *See, e.g.*, Sumihiro U.S. at 4:33-5:18, Fig. 4.
[1.1] a sorting unit for receiving memory requests and sorting said memory requests based on their addresses, wherein said memory requests are tagged for indicating a sending order thereof before said memory requests are sent to said sorting unit; and	MPC105 The MPC105 tagged memory request to indicate a sending order. Wang discloses tagging memory requests to indicate a sending order. *See, e.g.*, Wang at 44 ("All transactions entering the MPC105 have their addresses stored in buffers, which allows the MPC105 to snoop all transactions attempting to go through the device. This is important, for example, with a posted write operation that may execute out of order with other transactions."), Fig. 3. Stream Memory Controller Wulf discloses sorting memory requests based on address. *See, e.g.*, Wulf at Figs. 1-3, 9:66-10:9. Wulf discloses tagging memory requests to indicate a sending order. *See, e.g., id.* at [57], 3:43-59, 7:32-38, 16:13-19:50, 20:1-40:60, 42:47-67, 43:6-44:3. McKee I discloses sorting memory requests based on address. *See, e.g.*, McKee I at 5-6, Fig. 1. McKee II discloses tagging memory requests to indicate a sending order. *See, e.g., id.* at 3. McKee II discloses sorting memory requests based on address. *See, e.g.*, McKee II at 7-8, Fig. 3. McKee II discloses tagging memory requests to indicate a sending order. *See, e.g., id.* at 2. McGee discloses tagging memory requests to indicate a sending order. *See, e.g.*, McGee at 462. CDC-6600 The CDC-6600 tagged memory requests. Thornton discloses tagging memory requests. *See, e.g.*, Thornton at 47-50 ("At the time of entry a set of tags is also entered, which fully identify the nature of that particular storage reference. . . . Shown in Figure 36 is a worst-case condition of references filling the hopper and

	being recirculated out of order."), Figs. 35-36. OTHER REFERENCES Nakagawa discloses tagging memory requests to indicate a sending order. *See, e.g.*, Nakagawa at 2:8-18, 10:56-58, 11:26-30, 12:37-46, 13:35-14:2. Nakagawa discloses sorting memory requests based on address. *See, e.g., id.* at 2:58-67, 5:56-58. Wilson discloses tagging memory requests to indicate a sending order. *See, e.g.*, Wilson at 1:59-61, 2:58-60, 3:67-4:4, 5:18-22, 7:30-33, 36-43, 49-55, 60-65, 10:3-51, 59-67, 11:34-37, 12:20-26, 65-66, 12:67-13:6, 14:25-26. Chen discloses tagging memory requests to indicate a sending order. *See, e.g., id.* at 22:26-37 ("Data may be returned to the requesting ports **302**, **306** and **308** in a different order than it was requested. The arbitration node **44** receives a set of tags with each load address and queues them for future reference. When data is returned from main memory **14**, the tags are re-attached to the corresponding data words and both data and tags are passed back to the requesting port. The processors **100** and external interface means **22** use the tags to route the data to its proper location. For the vector means **104** and the external interface means **22**, the proper location insures correct sequencing of operations."), 22:64-23:15, 23:28-32, 30:47-48, Figs. 21c, 21d. Miller discloses tagging memory requests to indicate a sending order. *See, e.g.*, Miller at 17:9-50, 51-59 ("Because tags are created in sequential order when requests are made, using the request tags to address locations in the destination buffers **220** or **240** ensures that data is always loaded into the buffers **220** or **240** in proper sequence, even if the data returns in an arbitrary order. Reading data from the buffer in sequential order therefore guarantees that data is returned to the destination in proper order."), 18:64-19:2, claim 9. Foster discloses tagging memory requests to indicate a sending order. *See, e.g.*, Foster at 9:1-7, 30:40-48, 35:1-8, 39:39-44. Foster discloses sorting memory requests based on address. *See, e.g., id.* 15:62-16:2. Mote '534 discloses tagging memory requests to indicate a sending order. *See, e.g.*, Mote '534 at 2:34-49, 3:9-11, 11:1-19:20, 19:50-53, 20:46-48, 20:54-67, 22:12-17, Figs. 9-11, Abstract. Mote '494 discloses tagging memory requests to indicate a sending order. *See, e.g.*, Mote '494 at 2:33-48, 3:8-10, 11:1-19:7, 19:24-28, Figs. 9-11, Abstract. Isobe discloses tagging memory requests to indicate a sending order and sorting memory requests based on address. *See, e.g., id.* at 2:7-10, 3:35-49, 10:30-43 ("a plurality of identifier transmission means for dividing said access requests issued from said plurality of access request control units into a plurality of groups based on the order of issuance from said

	access request control units wherein each identifier transmission means is associated with one of said access request units and divides said access requests from said associated access request unit, and wherein each identifier transmission means further adds access requests in each group with access request identifiers, and wherein each identifier transmission means further transmits a plurality of access requests with access request identifiers added thereto."), 11:41-50, 12:52-55, Abstract. Wagner discloses sorting memory requests based on address. *See, e.g.*, Wagner, claim 3.
[1.2] a throughput maximizing unit for processing said memory requests to the synchronous DRAM in response to scheduling which maximizes the use of data slots by the synchronous DRAM.	CDC-6600 The CDC-6600 processed memory requests in response to scheduling constraints of the synchronous memory which maximizes the use of data slots. Thornton discloses processing memory requests in response to scheduling constraints of the synchronous memory which maximizes the use of data slots. *See, e.g.*, Thornton at 16, 46-50 (Memory performance may be improved by the following methods: "2. Split primary storage and provide overlap conditions to hide portions of the storage cycle. 3. Interleave many banks of primary storage in order to increase the speed of 'burst' transfers between the processor and storage, and between levels of storage."), Fig. 35. MPC105 The MPC105 processed memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. Wang discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Wang at 45 ("Address pipelining significantly improves data throughput by allowing the chip to decode a new address while the current data transaction finishes."). Garcia discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Garcia at 409-10. *MPC105 Summary* discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, *MPC105 Summary* at 5. 82C302 The 82C302 processed memory requests in response to scheduling constraints which maximized the use of data slots. U.S. Patent No. 4,924,375 to Fung (May 8, 1990) ("Fung '375") discloses processing memory requests in response to scheduling

	constraints which maximizes the use of data slots. *See, e.g.*, Fung '375 at 1:32-48 ("If one seeks to read sequential locations which are on different rows, after the first read cycle, one must wait for the pre-charge portion of the cycle to be completed before one can execute another read operation. In order to avoid this problem, one can use a known technique termed interleaved memory. In a conventional interleaved memory, there is an 'even' memory bank and an 'odd' memory bank. Any two sequential memory locations are stored in different memory banks. If one reads two sequential memory locations, the operation is as follows: the first location is read in the first memory bank and while that bank is being precharged, the second location is read from the second memory bank. Thus, since two sequential locations will be in different banks, they can be read very quickly without waiting for an intervening pre-charge cycle."), 1:64-67 ("A still further object of the present invention is to provide a memory accessing scheme which has a high probability of avoiding the necessity of delaying memory access due to the need to wait through a pre-charge cycle."), 2:22-26 ("The present invention provides a memory organization scheme for a high-performance memory controller. The memory organization of the present invention combines page mode techniques and interleaving techniques to achieve high-performance."), 2:35-42 ("While directly sequential memory locations are accessed, the system of the present invention takes advantage of the page accessing capability of DRAM memories. Where accessing is more than 2K away, the present invention takes advantage of an interleave scheme. Thus, the present invention combines the advantages of page memory organization and interleaved memory organization."); 5:68-6:1 ("The circuitry shown in FIG. 3 operates on a page interleaved mode. In each page several columns can be accessed without waiting for a precharge cycle. Likewise, one can alternate pages and select columns from already accessed pages without waiting for a precharge cycle. This means that if, for example, a series of memory requests includes (a) a number of requests for instructions that are located in one section of memory and (b) a request for several pieces of data that are located in a second section of memory, where the two memory sections are more than 2K apart, the data and the instructions can be accessed without waiting for a precharge cycle. The possible page organizations for the memory are shown in FIG. 4. As shown in FIG. 4, even and odd pages are located on different memory banks."), 6:14-7:17; *see also* Lin at 1:64-2:28. U.S. Patent No. 4,899,272 to Fung (Feb. 6, 1990) ("Fung '272") discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Fung '272 at 4:64-5:4, 5:34-50, 55-64. *AT386 Specification* discloses processing memory requests in response

	to scheduling constraints which maximizes the use of data slots. *See, e.g., AT386 Specification* § 6.0 ("The 82C302 will utilize either interleaving or page mode access directly to a DRAM array."), § 6.1. *82C302 Data Sheet* discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g., 82C302 Data Sheet* at 43 ("Page mode access with interleaved memory banks achieves higher performance than conventional DRAM arrays. . . . The 82C302 performs the memory control functions in a 80386-based systems [sic] that utilizes page mode access DRAMs."), 44 ("The frequency of the next access being fast (same or alternate page vs. alternate address in interleaved mode) is significantly higher."). Stream Memory Controller The Stream Memory Controller processed memory requests in response to scheduling constraints which maximizes the use of data slots. Wulf discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Wulf at 1:16-19, 2:28-44, 3:43-62, 6:38-56, 7:58-63, 9:19-27, 51-52, 9:66-10:9, 12:66-13:5, 13:32-38, 16:13-19:50, 20:1-40:60, 42:8-17, 25-67, 43:6-44:11. McKee I discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, McKee I at 4-5. McKee II discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, McKee II at 1-9. McGee discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, McGee at 462-63. MT48LC2M8S1 The Micron Data Sheet discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Micron Data Sheet at 2-43 ("bank switching between the two internal memory banks in conjunction with the programmable BURST mode provides very high-speed performance."); Williams '639 discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Williams '639 at 1:43-45 ("The interleaving between open banks coupled with the high speed burst mode may, in many cases, provide a 'seamless' flow of data."). Williams '162 discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the

	use of data slots. *See, e.g.*, Williams '162 at 1:36-38. Williams '551 discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Williams '162 at 1:35-42. Williams '901 discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Williams '162 at 1:38-45. CS4031 The 84031 processed memory requests in response to scheduling constraints which maximized the use of data slots. *CS4031 Information* discloses processes memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g., CS4031 Information* at 1, 33, 47-48, 59-61, 65, 78-80, 82. OTHER REFERENCES Ware '024 discloses processes memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Ware '024 at 4:65-5:6, 8:10-10:16. Allan discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Allan at 3:8-14, 19-39, 45-52, 6:32-47, claim 10. Kelly discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See e.g.*, Kelley at 3:3-4:16. Farmwald '327 discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Farmwald '327 at 10:15-11:39. Lang discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Lang at Fig. 4 ("ANY OF THE 8 BANKS MAY BE SCHEDULED IN EACH SLOT. IF BANK 0 IS BEING ACCESSED, A REFRESH MAY BE SCHEDULED FOR AN IDLE BANK IN SLOT 0. THIS ENHANCES SYSTEM PERFORMANCE."). Busch discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Busch at 649 ("A method to control sequential or interleaved data being written into or read from Dynamic Random Access Memory (DRAM) modules is disclosed. The sequential or interleaved data in this disclosure is refered [sic] to as a data burst, one or more data bits in length. Advanced applications for DRAMs require quick access times and high data rates. This method achieves an improvement in data rate by dynamically adjusting the length of the data burst, using the high logic level of an input signal named BC or Burst Control."), 653 ("Fig. 3

	shows that interleaved data between BURST0 and BURST1 output registers can be accomplished in a seamless manner. . . . The input circuit is designed to interleave between two data input registers."), 654 ("Fig. 6 shows that interleaved data between BURST0 and BURST1 data input registers can be accomplished in a seamless manner."), Figs. 3, 6. Lemmon II discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Lemmon II at 1:62-2:4 ("[T]he present invention transfers write data from a CPU to a memory module in a plurality of data bursts interspaced by a preselected number of bus cycles which the memory modules use to absorb the issued data burst thereby reducing the buffer requirements of the memory module. The present invention utilizes the interspaced bus cycles to send pending read commands to another memory module to interleave read and write operations to thereby lower memory read latency and optimize the available bandwidth on the bus."); 2:14-47; 14:35-38 ("This interleaving of a read command during a lull in a write transaction helps reduce overall memory read latency and makes better use of the available memory and bus bandwidth."); 15:51-16:33; Fig. 6; *see also* Lemmon I at 8:30-21:28, Figs. 4A, 4B, 5A, 5B. Bursky discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Bursky at 45-46 ("Such interleaved systems are able to deliver uninterrupted data streams when switching from bank to bank without any dead time on the data bus. . . . Workstations built with the synchronous memories could run their main-memory subsystems almost 400% faster than with standard asynchronous DRAMS, eliminating the need for large static-RAM-based caches."), 48 ("To further improve bus bandwidth, the memory chip can begin the row-precharge procedure two bits (two clocks) before the burst is complete, without interrupting the data-output burst. Systems designer can then hide 67% of the row precharge when the system runs at 66.7 MHz. One clock cycle after the burst is complete, RAS can be brought low again, and a new cycle is started. Thus, a total random access operation at a clock frequency of 66.7 MHz requires eight clock cycles (120 ns), including the row precharge."). Nakagawa discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Nakagawa at 1:61-2:23, 5:5-21, 5:66-6:2, 9:23-27, 38-41, 48-53, 58-63. Wilson discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Wilson at 2:5-9, 3:60-66, 4:5-10, 5:18-25, 7:18-8:11, Fig. 1c.

	Chen discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Chen at 6:30-34, 15:60-16:4, 16:68-17:5, 22:64-23:6, 23:28-32, 30:47-48. Miller discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Miller at 4:15-21, 16:59-67, 25:16-23, claim 9. Iyer discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See* Iyer at 8:53-56, 14:38-43, 15:3-12, 16:65-67, 17:16-19. Bechtolsheim discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Bechtolsheim at 1:48-53, 5:44-58. Foster discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Foster at 5:25-33, 6:10-65, 7:28-34, 15:66-16:2, 19:41-20:12, Fig. 5B. House discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, House at 7:44-58, claim 8. Carnevale discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Carnevale at 63-66. Nguyen discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Nguyen at 1:30-43, 2:9-31, 3:66-4:19, 6:46-8:15, 8:40-45. Mote '534 discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Mote '534 at 1:28-40, 1:50-2:7, 2:34-49, 2:57-65, 3:5-4:61, 8:40-54, 9:63-10:11, 10:38-19:20, 19:58-20:6, 20:54-67, 22:12-17, Abstract. Mote '494 discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Mote '494 at 1:27-39, 1:49-2:6, 2:33-48, 2:56-64, 3:4-4:61, 8:40-54, 9:63-10:11, 10:38-19:7, 19:25-28, 20:7-23, Abstract. Isobe discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Isobe at 3:17-22 ("Another object of this invention is to provide storage control apparatus and method capable of receiving access requests, issued concurrently from a plurality of access request control units, every clock pitch to greatly improve ability to process the access requests by synchronizing them with each other."), 5:28-48, 8:67-9:4; *see also id.* at. Johnson '236 discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the

	use of data slots. *See, e.g.*, Johnson '236 at 1:20-22, 35-38. Lin discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Lin. at 1:33-63, 2:55-59, 3:51-57, 4:40-5:1, 6:3-24, 7:3-5, Fig. 4, Abstract. Steele discloses processing memory requests in response to scheduling constraints which maximizes the use of data slots. *See, e.g.*, Steele at 1:6-24, 2:39-68, 5:18-22, 6:29-7:59, claims 1-13, Figs. 10, 12-13, Abstract. Mayer U.S. discloses processing memory requests in response to scheduling constraints of which maximizes the use of data slots. *See, e.g.*, Mayer U.S. at 1:9-12, 1:27-31, 1:61-2:17, 3:38-42, 4:18-19, 5:1-14:34, Figs. 3-6, claims 1-2, 5-8, 10-11, 15, 20, Abstract. Mayer Euro discloses processing memory requests in response to scheduling constraints of which maximizes the use of data slots. *See, e.g.*, Mayer Euro at [57], 1:1-4, 1:20-25, 1:55-2:27, 4:6-11, 4:56-57, 5:52-17:44, Figs. 3-6, claims 1-2, 5-6, 9-10, 13-14. Configurations for Solid States Memories, JEDEC Standard No. 21-C, release 4 (Nov. 1993) ("JEDEC 21-C") showed that the second internal bank of the SDRAM could accept a command while the first bank was busy operating on a prior command. Thus, a memory controller for a JEDEC-compliant SDRAM could operate both SDRAM banks concurrently to increase throughput. *See, e.g.*, JEDEC 21-C, Table 2. Bowater discloses processing memory requests in response to scheduling constraints of which maximizes the use of data slots. *See, e.g.*, Bowater at 3:13-20, 7:3-8:19, 8:36-60, 9:1-40, 9:50-10:2, 10:13-68, 11:18-42, Figs. 5, 7-10. Robert Adams & Gregory Scavone, "Design a DRAM controller from the top down," *Electronic Design News*, pp. 183-88 (Apr. 27, 1989) ("Adams & Scavone") discloses processing memory requests in response to scheduling constraints of which maximizes the use of data slots. *See, e.g.*, Adams & Scavone at 188. Wagner discloses processing memory requests in response to scheduling constraints of which maximizes the use of data slots. *See, e.g.*, Wagner at 2:2-5, 8:5-25, 54-56. Ito discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Ito at 8:35-46, 10:11-24, Fig. 9. Sumihiro U.S. discloses processing memory requests in response to scheduling constraints of the synchronous DRAM which maximizes the use of data slots. *See, e.g.*, Sumihiro U.S. at 4:33-5:18, Fig. 4. U.S. Patent No. 5,034,917 to Bland (Jul. 23, 1991) ("Bland") discloses processing memory requests in response to scheduling constraints of

	which maximizes the use of data slots. *See, e.g.*, Bland at 4:13-4:40.
3. A controller for a synchronous DRAM comprising:	The citations for the preamble of claim 1 are incorporated by reference for the preamble of claim 3. U.S. Patent No. 5,371,772 to Al-Khairi (Dec. 6, 1994) ("Al-Khairi") discloses system in which a processor communicates with a memory. *See, e.g.*, Al-Khairi at 1:7-2:27, 3:22-65, Fig. 1. U.S. Patent No. 5,256,994 to Langendorf (Oct. 26, 1993) ("Langendorf") discloses providing control clock signals to computer components. *See, e.g.*, Langendorf at 1:11-22.
[3.1] a sorting unit for receiving memory requests and sorting said memory requests based on their address;	The citations for the sorting unit of claim 1 are incorporated by reference for the sorting unit of claim 3.
[3.2] a throughput maximizing unit for processing said memory requests to the synchronous DRAM in response to scheduling which maximizes the use of data slots by the synchronous DRAM; and	The citations for the throughput maximizing unit of claim 1 are incorporated by reference for the throughput maximizing unit of claim 3.
[3.3] a control block for receiving a controller clock signal and developing an SDRAM clock signal by dividing said controller clock signal with a programmable divisor value.	MPC105 The MPC105 was capable of controlling SDRAMs having multiple clock speeds. *See, e.g., MPC105 Summary* at 20; Wang at 46 ("A phase-locked loop synchronizes the processor/memory bus to the PCI bus running at either the same or twice the speed of the PCI bus. This clocking scheme allows the MPC105 to interface with PowerPC microprocessors operating at 20- to 66-MHz processor bus frequencies."), 48 ("a system operating in a 2:1 clock mode where the processor/memory bus operates at 66 MHz and the PCI bus at 33 MHz"; "The internal clock can run either at the same frequency as the system clock or at twice the frequency of the input system clock."); Garcia at 409. The MPC105 had an external clock (SYSCLK) and an internal clock, which was either twice the frequency of the external clock or the same frequency as the external clock. *See, e.g., MPC105*

	Summary at 20. CS4031 The 84031 developed a DRAM clock signal by dividing a controller clock signal with a programmable divisor value. *CS4031 Information* discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g., CS4031 Information* at 17, 23, 32 (divisor value programmable to values of 1, 3, 4, 6, 8, 10, 12, 16, and 20), 45, 47, 51, 53-54, 77. JOHNSON/AMD TECHNOLOGY Johnson '912 discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. Johnson '912 at 2:45-67, 3:8-20, 4:60-63, 5:3-42, 6:53-7:9, 8:26-9:34, 9:52-10:32, 17:26-18:29. Johnson '869 discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Johnson '869 at 2:46-67, 3:8-21, 4:60-63, 5:3-42, 6:53-7:9, 8:26-9:34, 9:52-10:32, 17:26-18:30. OTHER REFERENCES Lemmon II discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Lemmon II at 3:52-55, 4:25-39. Lemmon I discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Lemmon I at 4:15-16, 5:10-24. U.S. Patent No. 5,477,181 to Li (Dec. 19, 1995) ("Li") discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Li at 1:11-13 ("Many types of frequency dividers are well known in the art, including those that are programmable for allowing the frequency division ratio to be selected and varied."),1:36-2:10, 2:18-23, 3:8-28, Figs. 1-2, Abstract (disclosed clock divider can be used "to achieve virtually any desired frequency division ratio"). Al-Khairi discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Al-Khairi at 1:9-12, 2:21-36, 3:39-42, 4:5-8, Fig. 2. Langendorf discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Langendorf at 2:12-21, 3:18-25, 3:31-44, 5:58-6:51, Fig. 1. Mayer U.S. discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.,* Mayer U.S. at 14:35-15:68, 18:6-29, Figs. 7, 10-11, 18, 20. Mayer Euro discloses developing a clock signal by dividing a controller

	clock signal with a programmable divisor value. *See, e.g.*, Mayer Euro at 18:2-19:54, 22:39-41, Figs. 7, 10-11, 18, 20. Wagner discloses developing a clock signal by dividing a controller clock signal with a programmable divisor value. *See, e.g.*, Wagner at 3:62-5:19.
10. A system for interfacing a processing device with a synchronous DRAM comprising:	The citations for the preamble of claim 1 are incorporated by reference for the preamble of claim 10.
[10.1] means for developing memory requests from the processing device;	The citations for the sorting unit of claim 1 are incorporated by reference for the means for developing memory requests from the processing device of claim 10.
[10.2] means for tagging said memory requests to indicate the order in which they are provided by the processing device; and	The citations for tagging in claim 1 are incorporated by reference for the means for tagging in claim 10.
[10.3] a controller for maximizing throughput of said memory requests from the processing device to the synchronous DRAM based on scheduling constraints of the synchronous DRAM and arbitrating between conflicting memory requests so that data slots used by the synchronous DRAM are maximized.	The citations for throughput maximizing in claim 1 are incorporated by reference for the controller for maximizing throughput in claim 10. <u>CDC-6600</u> The CDC-6600 arbitrated between conflicting memory requests. Thornton discloses arbitrating between conflicting memory requests. *See, e.g.*, Thornton at 48-49. <u>MPC105</u> The MPC105 arbitrated between conflicting memory requests. *See, e.g., MPC105 Summary* at 3 ("Major features" include "[a]rbitration for secondary processor on-chip."), 5; Wang at 44 (The MPC105 "compares the address of an incoming transaction, arbitrates for the shared data bus between the processor and memory, and controls the dataflow between the processor, PCI, and memory interfaces."); Garcia at 411-12. <u>Stream Memory Controller</u> Wulf discloses arbitrating between conflicting memory requests. *See, e.g.*, Wulf at 16:13-19:50, 20:1-40:60. McKee I discloses arbitrating between conflicting memory requests. *See, e.g.*, McKee I at 4-5.

	McKee II discloses arbitrating between conflicting memory requests. *See, e.g.*, McKee II at 2-9. McGee discloses arbitrating between conflicting memory requests. *See, e.g.*, McGee at 462-63. MT48LC2M8S1 The Mircon Data Sheet discloses arbitrating between conflicting memory requests. *See, e.g.*, Micron Data Sheet at 2-49 - 2-83. CS4031 The CS4031 arbitrated between conflicting memory requests. *CS4031 Information* discloses arbitrating between conflicting memory requests. *See, e.g.*, *CS4031 Information* at 47, 78-80. OTHER REFERENCES Lemmon II discloses arbitrating between conflicting memory requests. *See, e.g.*, Lemmon II at 5:49-67 Lemmon I discloses arbitrating between conflicting memory requests. *See, e.g.*, Lemmon I at 6:39-59. Farmwald '327 discloses arbitrating between conflicting memory requests. *See, e.g.*, *id.* at 12:46-14:49. Wilson discloses arbitrating between conflicting memory requests. *See, e.g.*, Wilson at 10:59-65, 12:29-13:6, Fig. 5a. Chen discloses arbitrating between conflicting memory requests. *See, e.g.*, *id.* at 17:67-18:2. Miller discloses arbitrating between conflicting memory requests. *See, e.g.*, *id.* at 14:28-41, 16:29-50, 25:40-47, claims 5-6. Nakagawa discloses arbitrating between conflicting memory requests. *See, e.g.*, Nakagawa at 3:33-39, 52-55, 4:5-9, 5:46-50, 6:47-55, 7:12-27, 12:42-46. Iyer discloses arbitrating between conflicting memory addresses. *See, e.g.*, *id.* at 3:13-19, 30-37, 9:39-51, 10:46-48, 13:3-15:49, 16:10-18:26, Figs. 2-3. Foster discloses arbitrating between conflicting memory requests. *See, e.g.*, Foster at 9:57-10:30. House discloses arbitrating between conflicting memory requests. *See, e.g.*, House at 1:41-45, 2:3-3:62, 7:61-8:14, 13:42-53, 14:46-51, Figs. 3-6, Abstract. Nguyen discloses arbitrating between conflicting memory requests. *See, e.g.*, Nguyen at 4:33-44, 8:38-39. Mote '534 discloses arbitrating between conflicting memory requests. *See, e.g.*, Mote '534 at 1:28-40, 1:50-2:7, 2:34-49, 3:5-4:61, 10:38-

	19:20, 19:58-20:6, 20:54-67, 22:12-17, Abstract. Mote '494 discloses arbitrating between conflicting memory requests. *See, e.g.*, Mote '494 at 1:27-39, 1:49-2:6, 2:33-48, 3:4-4:61, 10:38-19:7, 20:7-23, Abstract. Isobe discloses arbitrating between conflicting memory requests. *See, e.g.*, Isobe at 1:27-2:11, 5:28-43, 10:44-5, 11:29-40. Lin discloses arbitrating between conflicting memory requests. *See, e.g.*, Lin at 1:13-16, 2:39-44, 3:39-43, 4:17-5:20, 6:25-68, 7:5-10, Abstract. Mayer U.S. discloses arbitrating between conflicting memory requests. *See, e.g.*, Mayer U.S. at 4:50-61. Mayer Euro discloses arbitrating between conflicting memory requests. *See, e.g.*, Mayer Euro at 5:33-44. Adams & Scavone discloses arbitrating between conflicting memory requests. *See, e.g.*, Adams & Scavone at 184, Figs. 2-4, A.
11. A method for controlling a synchronous DRAM comprising the steps of:	The citations for a controller in claim 1 are incorporated by reference for the method for controlling in claim 11.
(a) receiving memory requests and sorting said memory requests based on their addresses;	The citations for sorting in claim 1 are incorporated by reference for the step of sorting in claim 11.
(b) tagging said memory requests to indicate a sending order thereof before said memory requests are received at said step (a); and	The citations for tagging in claim 1 are incorporated by reference for the step of tagging in claim 11.

(c) maximizing throughput of said memory requests to the synchronous DRAM so that use of data slots by the synchronous DRAM is maximized.	The citations for maximizing throughput in claim 1 are incorporated by reference for the step of maximizing throughput in claim 11.
13. A method for controlling a synchronous DRAM comprising the steps of:	The citations for a controller in claim 1 are incorporated by reference for the method of controlling in claim 13.
(a) receiving memory requests and sorting said memory requests based on their addresses;	The citations for sorting in claim 1 are incorporated by reference for the step of sorting in claim 13.
(b) maximizing throughput of said memory requests to the synchronous DRAM so that use of data slots by the synchronous DRAM is maximized; and	The citations for maximizing throughput in claim 1 are incorporated by reference for the step of maximizing throughput in claim 13.
(c) receiving a controller clock signal and developing an SDRAM clock signal by dividing said controller clock signal with a programmable divisor value.	The citations for a programmable clock divisor in claim 1 are incorporated by reference for the step of developing an SDRAM clock signal by dividing said controller clock signal with a programmable divisor value in claim 13.

20. A method for interfacing a processing device with a synchronous DRAM, comprising the steps of:	The citations for a controller in claim 1 are incorporated by reference for the method for interfacing a processing device in claim 20.
(a) developing memory requests from the processing device;	The citations for sorting in claim 1 are incorporated by reference for the step of developing memory requests in claim 20.
(b) tagging said memory requests to indicate the order in which they are provided by the processing device; and	The citations for tagging in claim 1 are incorporated by reference for the step of tagging in claim 20.
(c) maximizing throughput of said memory requests from the processing device to the synchronous DRAM based on scheduling constraints of the synchronous DRAM and arbitrating between conflicting memory requests so that the data slots used by the synchronous DRAM are maximized.	The citations for maximizing throughput in claim 1 are incorporated by reference for the step of maximizing throughput in claim 20. The citations for arbitrating in claim 10 are incorporated by reference for the step of arbitrating in claim 20.